

42390P9770

SEP 03 2009

PATENT

AMENDMENTS TO THE CLAIMS

- 1-8. (Cancelled)
9. (Currently Amended) A method comprising:
running guest software in a processor mode that enables the guest software to
operate at a privilege level intended by the guest software;
identifying, within said processor mode, an attempt of the guest software to
perform an operation restricted by said processor mode;
determining, within said processor mode, whether the attempt of the guest
software would fail if the guest software was running outside said processor
mode;
allowing the guest software to attempt the operation within said processor mode
in response to determining that the attempt would fail if the guest software
was running outside said processor mode; and
transferring control over the operation to an operating system running within said
processor mode in response to the guest software attempting the operation.
- 10-30. (Canceled)
31. (Previously Presented) The method of claim 9 wherein determining that the
attempt of the guest software would fail includes determining that the guest software
is running with insufficient privilege to perform the operation.
32. (Previously Presented) The method of claim 9 further comprising exiting said
processor mode to transfer control over the operation to a virtual machine monitor
running outside said processor mode in response to determining that the attempt
would succeed if the guest software was running outside said processor mode.
33. (Currently Amended) A processor comprising:
a storage location to store an indicator to indicate whether the processor is

42390P9770

PATENT

configured to execute guest software in a mode that enables the guest software to operate at a privilege level intended by the guest software; and logic to execute the guest software in said processor mode, to identify, within said processor mode, an attempt of the guest software to perform an operation restricted by said processor mode, to determine, within said processor mode, whether the attempt would fail if the guest software was running outside said processor mode, to allow the guest software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode, and to transfer control over the operation to an operating system running within said processor mode in response to the attempt.

34. (Previously Presented) The processor of claim 33 wherein the logic is to determine whether the attempt would fail by determining whether the guest software is running with sufficient privilege to perform the operation.
35. (Previously Presented) The processor of claim 33 wherein the logic is also to exit said processor mode to transfer control over the operation to a virtual machine monitor running outside said processor mode in response to determining that the attempt would succeed if the guest software was running outside said processor mode.
36. (Currently Amended) A system comprising:
a memory to store application software and an operating system; and
a processor to execute the application software in a processor mode that enables the application software to operate at a privilege level intended by the application software, to identify, within said processor mode, an attempt of the application software to perform an operation restricted by said processor mode, to determine, within said processor mode, whether the attempt would fail if the application software was running outside said processor mode, to allow the application software to attempt the operation within said processor

42390P9770

PATENT

mode in response to determining that the attempt would fail if the guest software was running outside said processor mode, and to transfer control over the operation to the operating system running within said processor mode in response to the attempt.

37. (Previously Presented) The system of claim 36 wherein the processor is to determine whether the attempt would fail by determining whether the application software is running with sufficient privilege to perform the operation.
38. (Previously Presented) The system of claim 36 wherein:
the memory is also to store a virtual machine monitor; and
the processor is also to exit said processor mode to transfer control over the operation to the virtual machine monitor running outside said processor mode in response to determining that the attempt would succeed.
39. (Currently Amended) A computer readable storage medium that provides instructions, which when executed on a processor, cause the processor to:
present a virtual machine to guest software in a processor mode that enables the guest software to operate at a privilege level intended by the guest software;
and
handle a virtualization trap, where the virtualization trap is generated in response to an attempt of the guest software to perform an operation restricted by said processor mode and a determination within said processor mode, that the attempt would succeed if the guest software was running outside the virtual machine.
40. (Previously Presented) The computer readable storage medium of claim 39 wherein the determination that the attempt would succeed includes determining whether the application software is running with sufficient privilege to perform the operation.